Fortnightly eFolio Submission – Studio 7

Student Name and ID
Hao Xu (32767919)

Self-Evaluation
High Distinction

Task 7.1 (Topic Presentation and ASP.NET Identity)

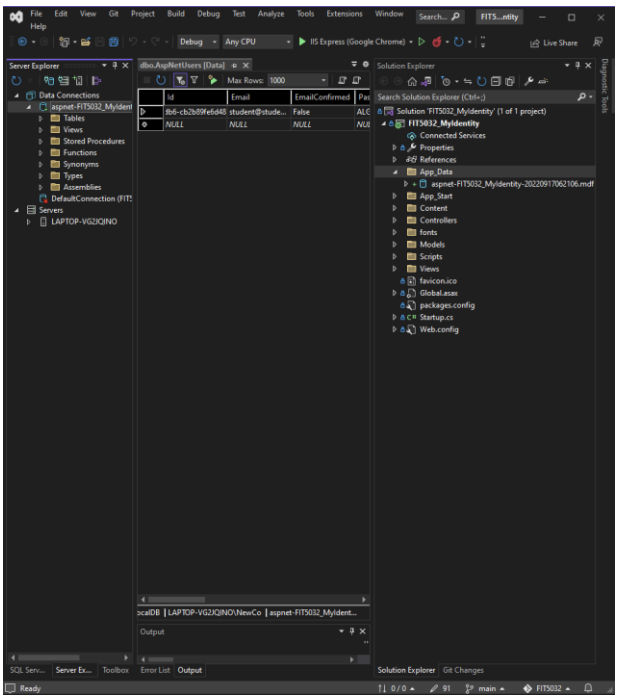
Topic Presentation (MD5 and SHA1's weaknesses and alternative)
In this tutorial, the topic related to MD5 and SHA1 is assigned to our group. I was responsible for researching the weaknesses of MD5. There are several aspects of presentation that can be improved. For example, the history of MD5, the details and functions of MD5 can be included. Also, some presentation technics can be enhanced, such as introducing presenters' name, introducing the next presenter, etc.

The presentation slide is included in the GitHub Link. Also, the slides are included in the last pages of this document as appendix.

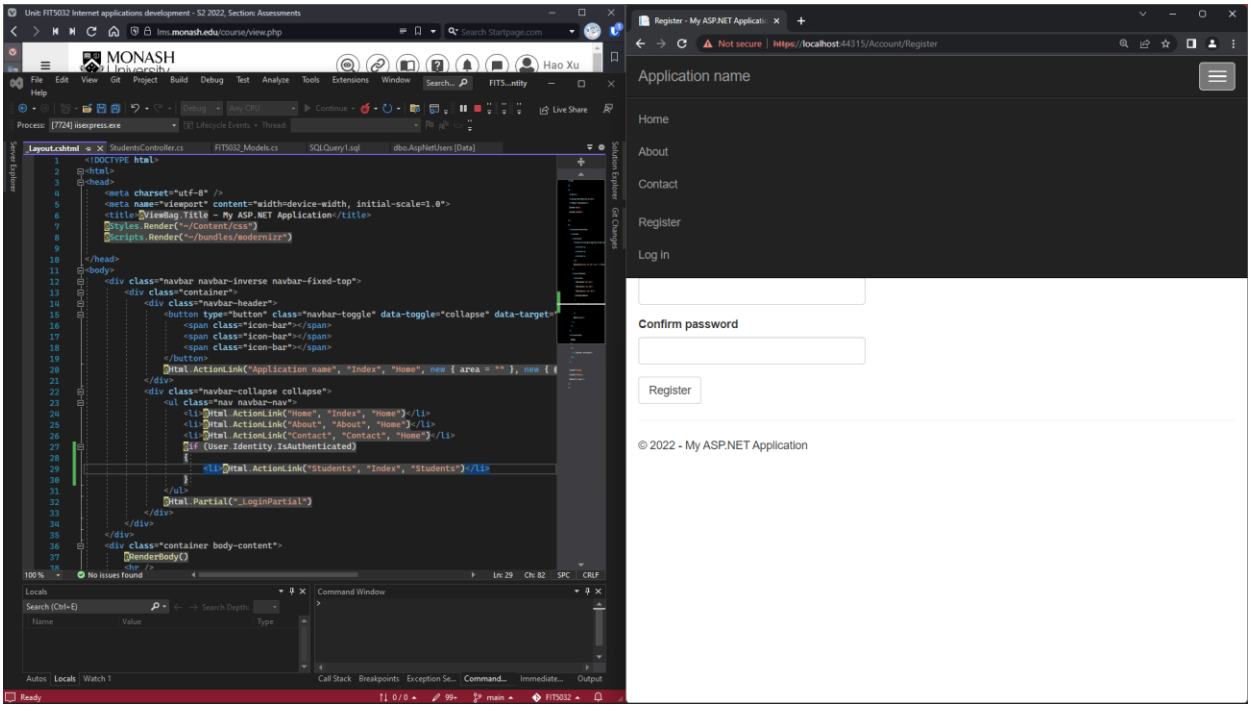
GitHub Link: https://github.com/BrantleyXU/FIT5032/blob/main/week07/FIT5032-Week07-Group1.pptx

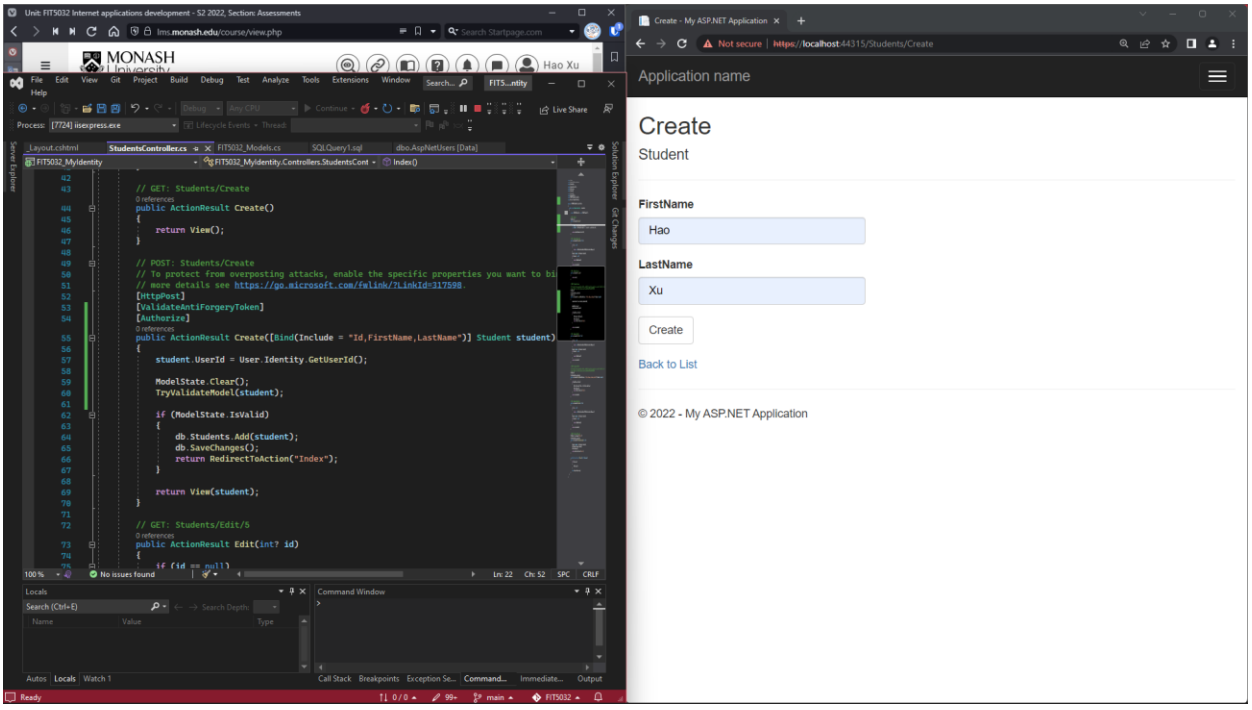
ASP.NET Identity
GitHub Link:
https://github.com/BrantleyXU/FIT5032/tree/main/week07/FIT5032_MyIdentity

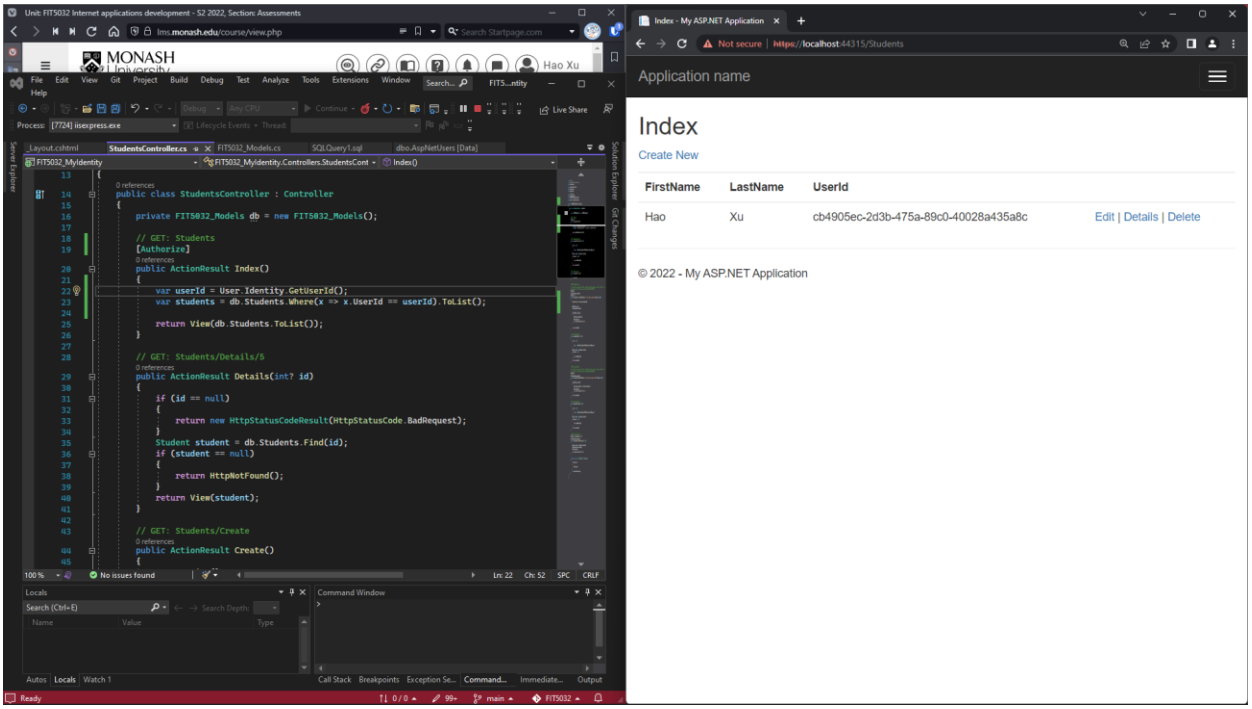
The following screenshot shows that a user is successfully created.



The following screenshots show that MS Identity is successfully setup.



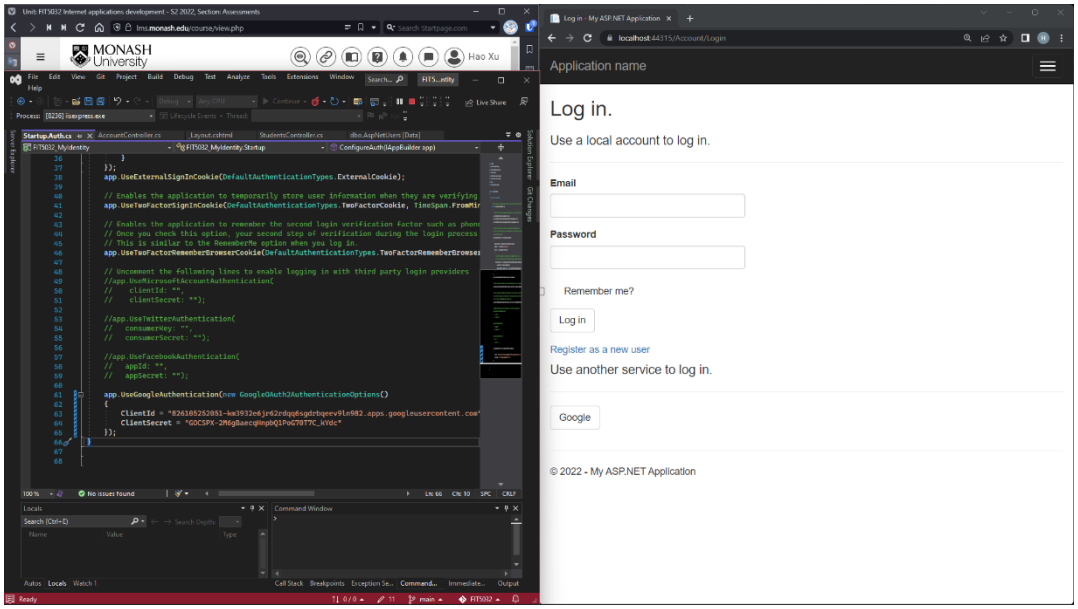


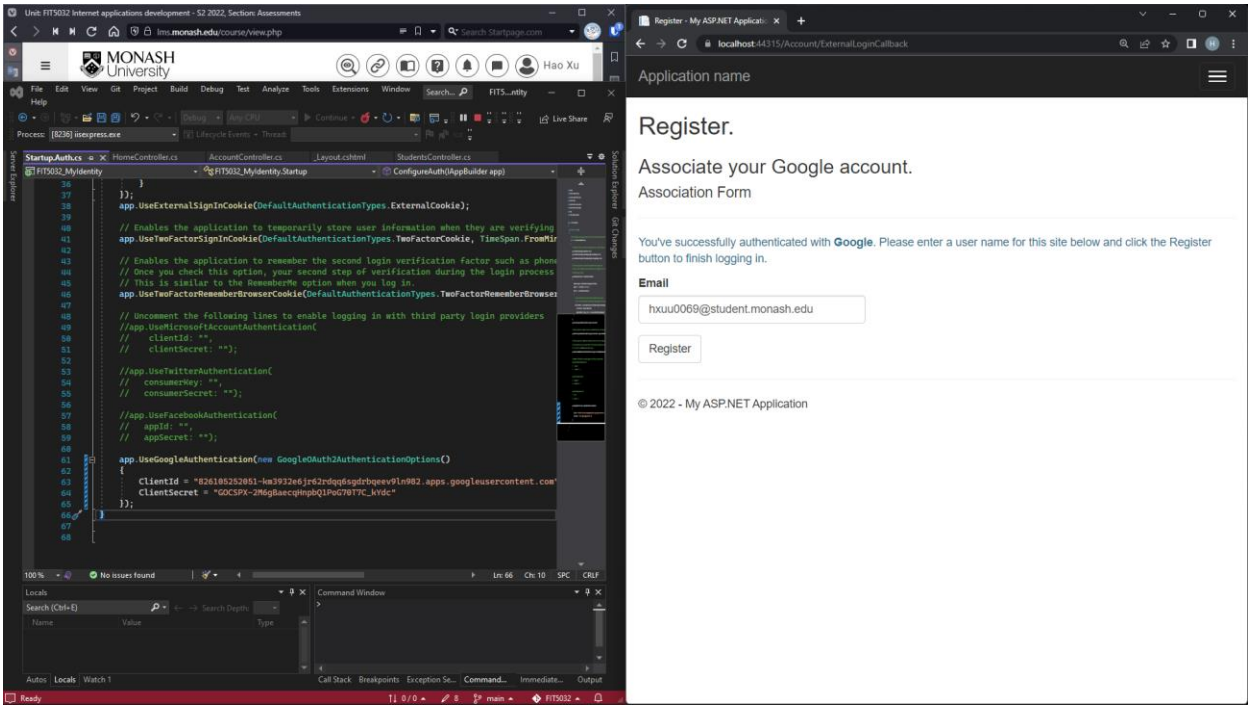


Task 7.2 (MS Identity with Google Authentication)

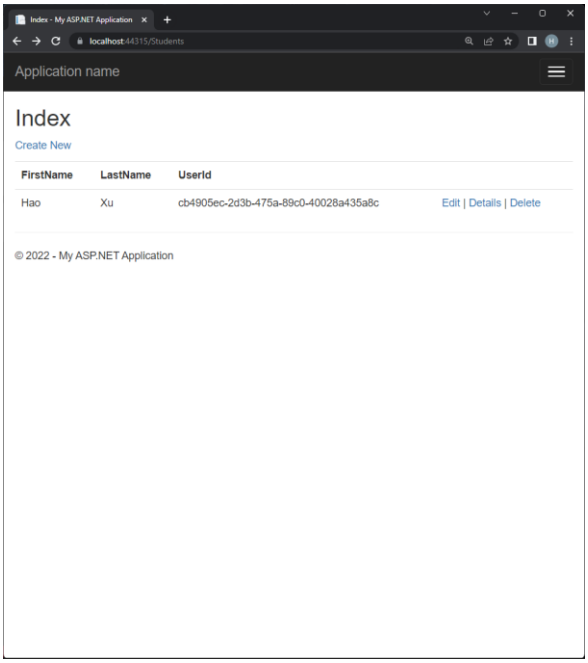
GitHub Link of "Startup.Auth.cs" file:
https://github.com/BrantleyXU/FIT5032/blob/main/week07/FIT5032_MyIdentity/FIT5032_MyIdentity/App_Start/Startup.Auth.cs

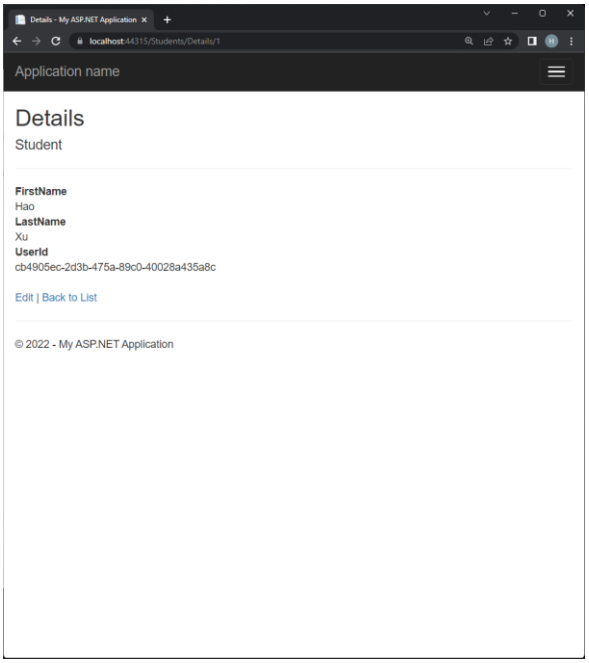
The following screenshots show that Google Authentication is successfully integrated with the application. Accounts registration can be done via Google Authentication. The "Startup.Auth.cs" file is also shown in the screenshots.



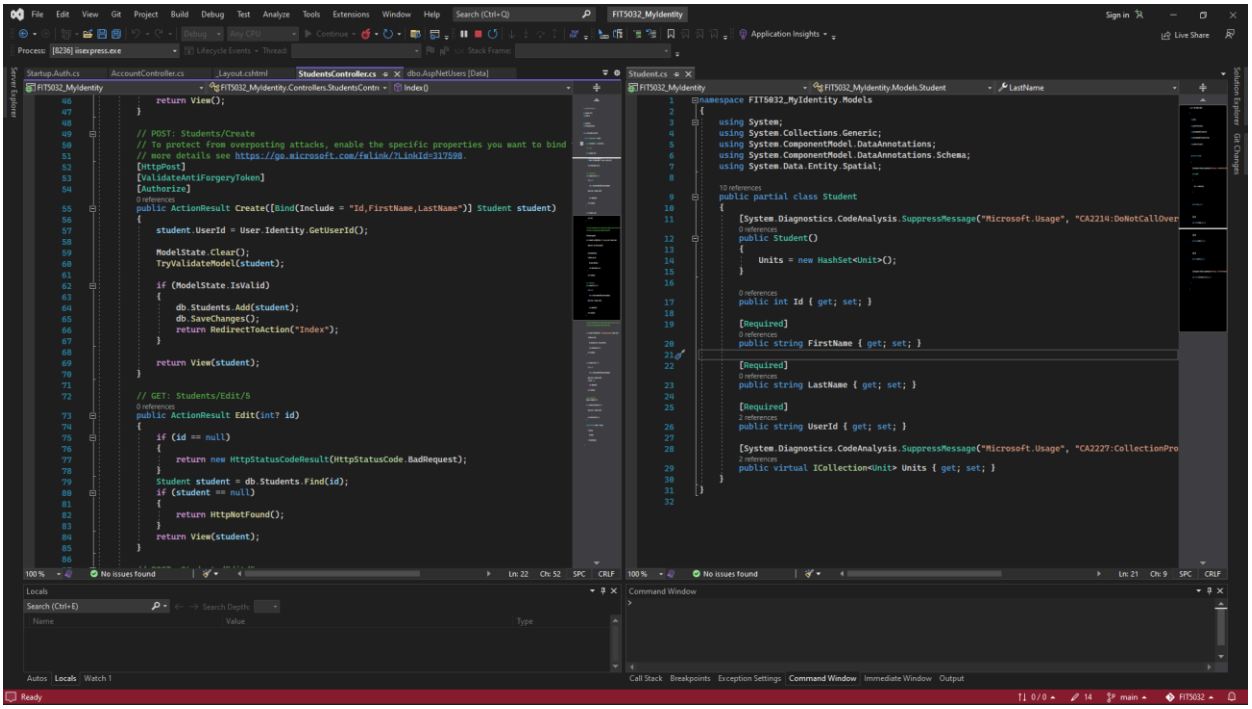


The following screenshots show that the list of students and students' details.





The following screenshot shows that the codes of student controller and student model.


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
FIT5032_MyIdentity
Sign in
Live Share
Application Insights
Process: [8236] iisexpress.exe
Startup.Auth.cs
AccountController.cs
_Layout.cshtml
StudentsController.cs
dbo.AspNetUsers [Data]
FIT5032_MyIdentity.Controllers.StudentsContr
Index()
return View();
}
// POST: Students/Create
// To protect from overposting attacks, enable the specific properties you want to bind
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
[Authorize]
0 references
public ActionResult Create([Bind(Include = "Id,FirstName,LastName")] Student student)
{
student.UserId = User.Identity.GetUserId();
ModelState.Clear();
TryValidateModel(student);
if (ModelState.IsValid)
{
db.Students.Add(student);
db.SaveChanges();
return RedirectToAction("Index");
}
return View(student);
}
// GET: Students/Edit/5
0 references
public ActionResult Edit(int? id)
{
if (id == null)
{
return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
}
Student student = db.Students.Find(id);
if (student == null)
{
return HttpNotFound();
}
return View(student);
}
No issues found
Ln: 22 Ch: 52 SPC CRLF
Student.cs
FIT5032_MyIdentity.Models.Student
LastName
namespace FIT5032_MyIdentity.Models
{
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System.Data.Entity.Spatial;
10 references
public partial class Student
{
[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOver
0 references
public Student()
{
Units = new HashSet<Unit>();
}
0 references
public int Id { get; set; }
[Required]
0 references
public string FirstName { get; set; }
[Required]
0 references
public string LastName { get; set; }
[Required]
2 references
public string UserId { get; set; }
[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPro
2 references
public virtual ICollection<Unit> Units { get; set; }
}
}
No issues found
Ln: 21 Ch: 9 SPC CRLF
Locals
Search (Ctrl+E)
Search Depth:
Name
Value
Type
Autos Locals Watch 1
Command Window
Call Stack Breakpoints Exception Settings Command Window Immediate Window Output
Ready
main
FIT5032

MD5 and SHA-1

Weakness and Alternative

HAO XU (32767919)
HAOCHEN LI (28619773)
KAI LI (32746741)

1

MD5 (Message-Digest Algorithm 5)

- A cryptographic algorithm, a one-way hash function, often used to store passwords in a database
- that produce a 128-bit hash value (32 characters) string from any password, phrase or text
- not secure any more.

- Weakness:
- Brute force attacks on MD5 hashes are fast:
  - The MD5 algorithm is fast to use, so in a few seconds the attacker can try so many combinations.
  - An NVIDIA GeForce 8800 Ultra can calculate more than 200 million hashes per second
- MD5 has collisions:
  - A collision is when two words have the same hash generated. However, MD5 is proved to have a low collision resistance, meaning that the attacker may be able to produce the same hash for two different inputs.
  - A collision attack exists that can find collisions within seconds on a computer with a 2.6 GHz Pentium 4 processor

2

SHA1

SHA-1 stands for **Secure Hash Algorithm 1**, it was designed by the United States National Security Agency.

SHA-1 is one of several **cryptographic hash functions** which takes an input and produces a 160-bite (20-byte) hash value.

This hash value is known as a **message digest**. This message digest is usually then rendered as a hexadecimal number which is 40 digits long.

However, SHA-1 is now considered insecure since 2005. And major tech giants browsers like Microsoft, Google, Apple have stopped accepting SHA-1 SSL certificates by 2017.

3

SHA-1 Weakness

Collision

Security researchers have achieved the first real-world collision attack against the SHA-1 hash function, producing two different PDF files with the same SHA-1 signature

4

SHA-256 (alternative of MD5 and SHA - 1)

MD5 produces 32 chars hash, SHA-1 produces 40 chars hash, SHA-256 produces 64 chars hash which is harder to crack.

Use SHA-256 alternative because this 256-bit key is much more secure than other common hashing algorithms:

It's a secure and trusted industry standard:

SHA-256 is an industry standard that is trusted by leading public-sector agencies and used widely by technology leaders.

Collisions are incredibly unlikely:

There are SHA-256 possible hash values when using SHA-256, which makes it nearly impossible for two different documents to coincidentally have the exact same hash value.

The avalanche effect:

Unlike some older hashing algorithms, even a very minor change to the original information completely changes the hash value

5

Thanks for your listening!

6